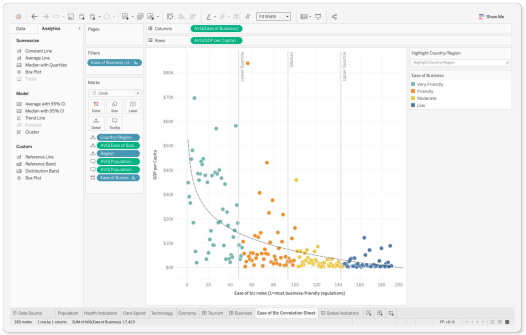
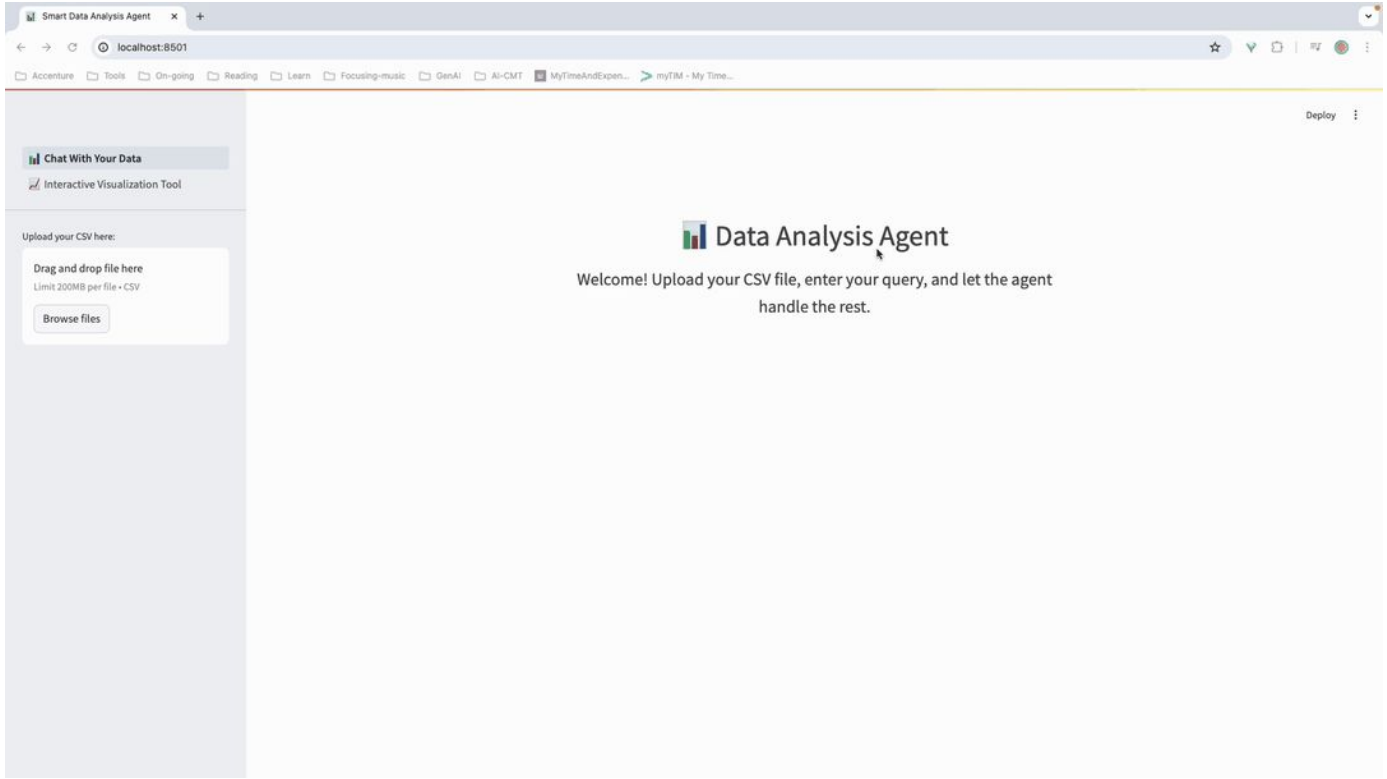
Creating a Custom Data Analysis Tool with LLMs - Easy Step-by-Step Guide

Data Science Lab | Your Guide to Data Science Mastery



1. Introduction to DAGPT - Demo



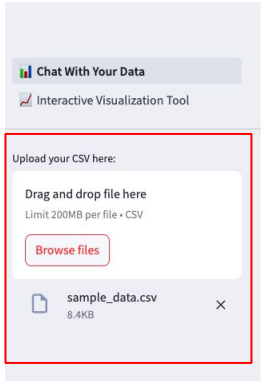
1. Introduction to DAGPT - Overview

- **DAGPT:** Data Analysis GPT is a helpful data analysis tool that uses LLMs to assist with data manipulation and analysis tasks via a conversational interface.
- Using **Streamlit** to provide an interactive web application where users can upload their data, ask questions, explore your data interactively and receive insights.
- **Alternative tool for Chat GPT-4o and Tableau** and suitable for any data analysis tasks in your local machine.

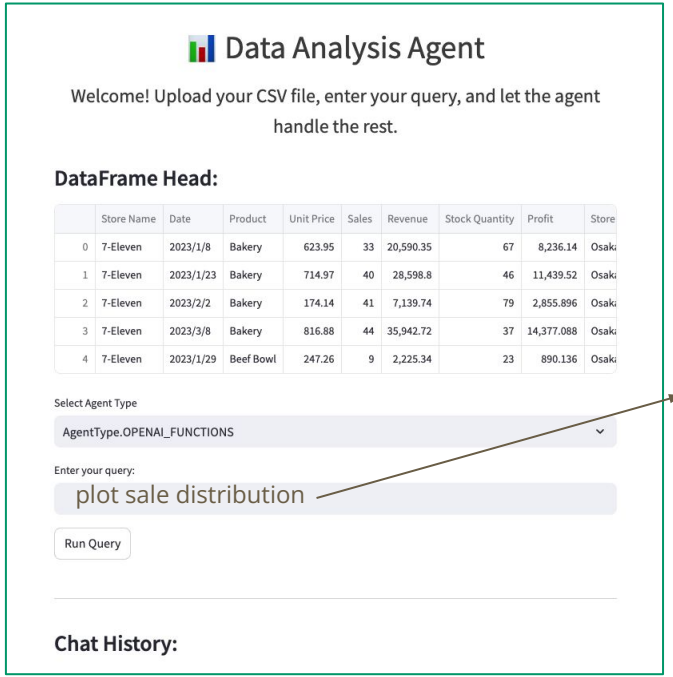
1. Introduction to DAGPT - Features (1/3)

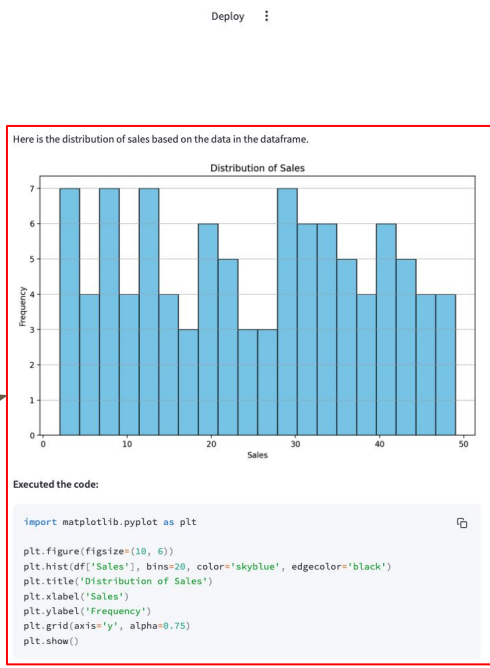
- **CSV File Upload:** Easily upload your CSV data through the sidebar.
- **Data Analysis:** Enter queries about your data and get responses powered by LLMs.
- **Data Visualization:** Generate and display plots based on your data queries.
- **Interactive Visualization Tool:** Explore your data interactively using various visualizations. Upload your dataset and start analyzing with drag-and-drop functionality to create customized charts.

1. Introduction to DAGPT - Features (2/3)



CSV File Upload: Easily upload your CSV data through the sidebar.

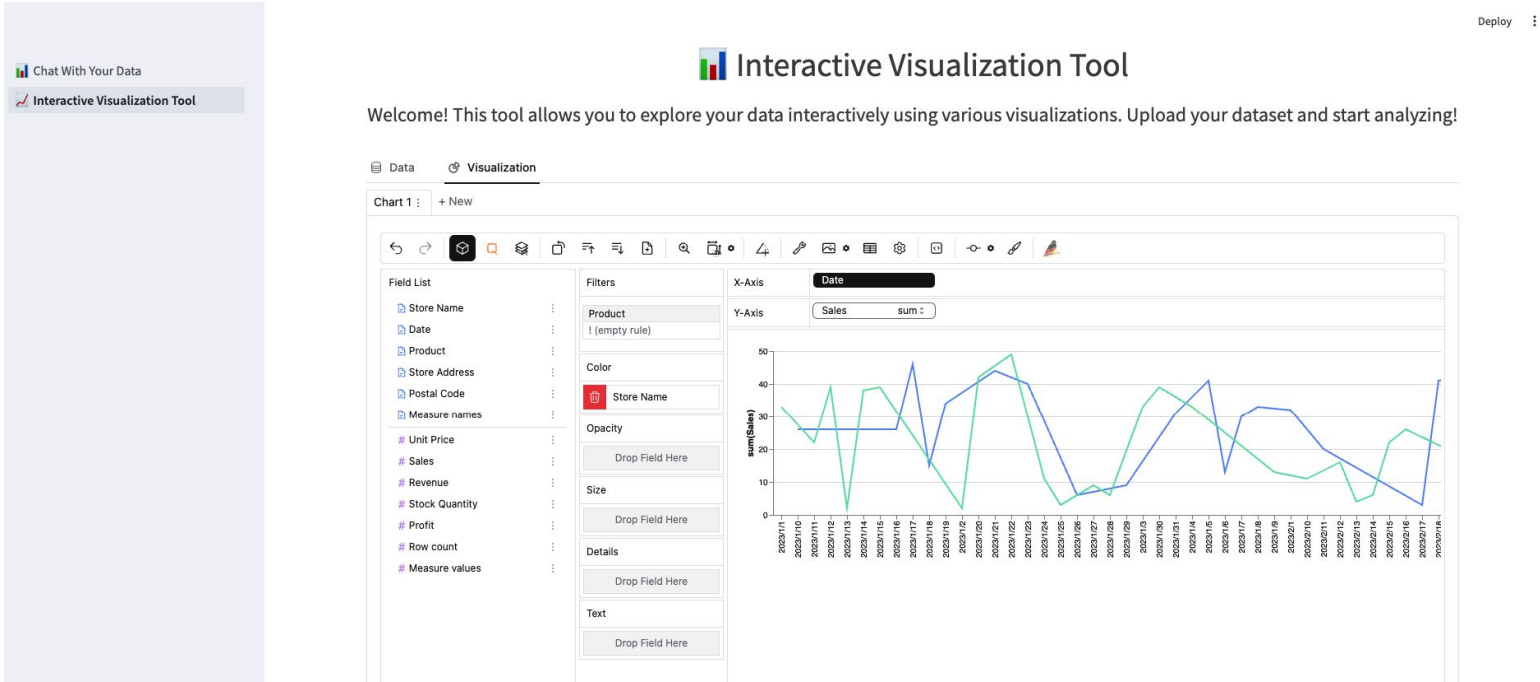




Data Analysis: Enter queries about your data and get responses powered by LLMs.

Data Visualization: Generate and display plots based on your data queries.

1. Introduction to DAGPT - Features (3/3)



Interactive Visualization Tool: Explore your data interactively using various visualizations. Upload csv data and start analyzing with drag-and-drop functionality to create customized charts.

2. Tools and Packages Needed to Build This App

Python 3.9 or Higher:

- The primary programming language used for developing the app.

Langchain:

- A framework for developing applications powered by language models. It provides tools to connect language models with other data sources, manage interactions, and create advanced AI applications.

Streamlit:

- An open-source app framework for Machine Learning and Data Science projects. It allows you to create interactive web apps with Python.

Pygwalker:

- A tool for creating interactive visualizations.

OpenAI API (optional):

- Used to access advanced language models for enhanced capabilities.

3. Tutorial - Steps

Step 1 | Step 2 | Step 3

Basic setup

- Build a project structure
- Setup virtual environment
- Install necessary packages
- Setup git reporitoty to store our project code

Build “Chat With Your Data” Tool

- Use streamlit to build our UI interface
- Use agent toolkits provided by langchain to build our data analysis agent
- Use OpenAI GPT model

Build “Interactive Visualization” Tool

- Embed pygwalker into our streamlit interface to build our interactive visualization tool

Link to github: https://github.com/datasciencelab-ai/dagpt

3. Tutorial - Basic setup

Project structure:

- Using “Cookiecutter Data Science” to build a logical standardized project structure

Setup virtual environment:

```
python -m venv .venv
source .venv/bin/activate   # On Windows use `.venv\Scripts\activate`
```

Install necessary packages: pip install -r requirements.txt

```
# requirements.txt
langchain==0.2.6
langchain-community==0.2.6
langchain-openai==0.1.13
streamlit==1.36.0
pygwalker==0.4.9.1
pandas==1.5.3
matplotlib==3.9.0
```

3. Tutorial - Live coding

```python
def main():

    st.set_page_config(page_title="Smart Data Analysis Tool", page_icon="📊", layout="centered")

    st.header("📊 Smart Data Analysis Tool")

    # Load model
    llm = load_llm(model_name=MODEL_NAME)
    logger.info(f"Succesfully loaded {MODEL_NAME} !")

    with st.sidebar:
        uploaded_file = st.file_uploader("Upload your csv file here: ", type="csv")

    if "history" not in st.session_state:
        st.session_state.history = []

    if uploaded_file is not None:
        st.session_state.df = pd.read_csv(uploaded_file)

        # display uploaded dataframe
        display_data(st.session_state.df)

        # Create our agent to help us answer all our question about our data
        da_agent = create_pandas_dataframe_agent(
            llm=llm,
            df=st.session_state.df,
            allow_dangerous_code=True,
            verbose=True,
            return_intermediate_steps=True,
            agent_type="tool-calling",
        )

        # Query
        query = st.text_input("Enter your question: ")

        if st.button("Run Query"):
            with st.spinner("Processing..."):
                process_query(da_agent, query)

        st.divider()
        display_chat_history()


if __name__ == "__main__":
    main()
```

4. Conclusion

- With the capability of LLMs, **it is very easy to write an app that do a daily complex data analysis tasks.**
- Using **Streamlit** help to easily build a conversational interface and powerful visualization capabilities, it makes data analysis accessible to all.
- **Pygwalker** is a free and useful tools alternative to Tableau and can be embedded in our own app
- By following this guide, **everyone can build your own data analysis tool leveraging the power of LLMs and Streamlit**

Appendix

How to setup OpenAI API key

Sign Up / Log In to OpenAI:

- Visit the OpenAI website and create an account or log in if you already have one.

Navigate to API Keys:

- Once logged in, go to the API section in your account dashboard. This is typically found under the "API Keys" tab.

Create a New API Key:

- Click on the "Create new secret key" button. This will generate a new API key for you to use.

Copy the API Key:

- After the key is generated, make sure to copy it and replace your API key in the `.env` file in your project.

```
1  OPENAI_API_KEY=your-openai-api-key
```

1. Introduction to DAGPT - Demo